National Information Assurance Partnership

TM

Common Criteria Evaluation and Validation Scheme Validation Report

HP

Red Hat Enterprise Linux

Version 5

Report Number: CCEVS-VR-07-0054

Dated: 2007-06-26

Version: 1.0

National Institute of Standards and Technology
Information Technology Laboratory
100 Bureau Drive
Gaithersburg, MD 20899

National Security Agency
Information Assurance Directorate
9800 Savage Road STE 6740
Fort George G. Meade, MD 20755-6740

ACKNOWLEDGEMENTS

Validation Team

The Aerospace Corporation

Columbia, MD

Noblis

Falls Church, VA

atsec Information Security Corporation

Austin, TX

Table of Contents

1. EXECUTIVE SUMMARY

This report documents the NIAP validators' assessment of the evaluation of HP Red Hat Enterprise Linux (RHEL) Version 5 Server and Red Hat Enterprise Linux Version 5 Client. It presents the evaluation results, their justifications, and the conformance results. This validation report is not an endorsement of the IT product by any agency of the U.S. Government and no warranty of the IT product is either expressed or implied.

The evaluation was performed by the atsec Information Security Corporation, and was completed during May 2007. atsec Information Security Corporation is an approved NIAP Common Criteria Testing Laboratory (CCTL). The evaluation was conducted in accordance with the requirements of the Common Criteria for Information Technology Security Evaluation, version 2.3. The information in this report is largely derived from the Evaluation Technical Report (ETR) and associated test report, both written by the CCTL. The evaluation determined the product to be **Part 2 extended, Part 3 conformant**, and to meet the requirements of **EAL4 augmented by ALC_FLR.3**. Additionally, the TOE was shown to satisfy the requirements of the Controlled Access Protection Profile (CAPP), Version 1.d, 8 October 1999, Labeled Security Protection Profile (LSPP), issue 1.b, 8 October 1999, and Role-based Access Control Protection Profile, Version 1.0, July 30, 1998.

Red Hat Enterprise Linux is a general purpose, multi-user, multi-tasking Linux based operating system. It provides a platform for a variety of applications in the governmental and commercial environment. Red Hat Enterprise Linux is available on a broad range of computer systems, ranging from departmental servers to multi-processor enterprise servers and small server type computer systems.

The Red Hat Enterprise Linux evaluation covers a potentially distributed, but closed network of HP (Itanium2, Pentium, Xeon, and Opteron based) servers running the evaluated version of Red Hat Enterprise Linux. The hardware platforms selected for the evaluation consist of machines which are available when the evaluation has completed and to remain available for a substantial period of time afterwards.

The validation team agrees that the CCTL presented appropriate rationales to support the Results of Evaluation presented in Section 4, and the Conclusions presented in Section 5 of the ETR. The validation team therefore concludes that the evaluation and the Pass results for Red Hat Enterprise Linux v5 is complete and correct.

2. IDENTIFICATION

The Common Criteria Evaluation and Validation Scheme (CCEVS) is a National Security Agency (NSA) effort to establish commercial facilities to perform trusted product evaluations. Under this program, security evaluations are conducted by commercial testing laboratories called Common Criteria Testing Laboratories (CCTLs) using the Common Evaluation Methodology (CEM) for Evaluation Assurance Level (EAL) 1 through EAL 4 in accordance with National Voluntary

Laboratory Assessment Program (NVLAP) accreditation granted by the National Institute of Standards and Technology (NIST).

The NIAP Validation Body assigns Validators to monitor the CCTLs to ensure quality and consistency across evaluations. Developers of information technology products desiring a security evaluation contract with a CCTL and pay a fee for their product's evaluation. Upon successful completion of the evaluation, the product is added to NIAP's Validated Products List.

Table 1 provides information needed to completely identify the product, including:

- The Target of Evaluation (TOE): the fully qualified identifier of the product as evaluated;
- The Security Target (ST), describing the security features, claims, and assurances of the product;
- The conformance result of the evaluation;
- The Protection Profile to which the product is conformant;
- The organizations and individuals participating in the evaluation.

Table 1: Evaluation Identifiers

Item	Identifier
Evaluation Scheme	United States NIAP Common Criteria Evaluation and Validation Scheme
Target of Evaluation	HP Red Hat Enterprise Linux, Version 5 Server HP Red Hat Enterprise Linux, Version 5 Client
Protection Profile	Controlled Access Protection Profile (CAPP), Issue 1.d, 8 October 1999. Labeled Security Protection Profile (LSPP), issue 1.b, 8 October 1999 Role-based Access Control Protection Profile, Version 1.0, July 30, 1998.
Security Target	*Red Hat Enterprise Linux Version 5 Security Target for CAPP,LSPP and RBAC compliance;* Version 3.9, 31 May 2007
Evaluation Technical Report	*Evaluation Technical Report a Target of Evaluation: Red Hat Enterprise Linux Version 5* Version 1.0, 31 May 2007
Conformance Result	CC V2.3, Part 2 extended, Part 3 conformant, EAL 4 augmented by ALC_FLR.3, and CAPP/LSPP/RBAC-compliant
Sponsor	HP
Developer	HP and Red Hat
Evaluators	Atsec information security corporation
Validators	The Aerospace Corporation Noblis

3. SECURITY POLICY

3.1. Discretionary Access Control

Red Hat Enterprise Linux implements Discretionary Access Control (DAC) through the use of standard UNIX permission bits and the POSIX standard Access Control Lists (ACLs). A Discretionary Access Control policy requires mechanisms whereby the access of users (i.e., subjects) to system resources and data (i.e., objects) can be controlled on the basis of user identity, role, and explicit permissions. Mechanisms that implement a DAC policy provide the capability for users to specify the how their personal data objects are to be shared.

Permission bits are associated with objects and specify the permissions (typically, READ, WRITE, EXECUTE) for a specific user, the user's group affiliation, and all others (i.e., "world"). Access Control Lists provide the same functionality relative to granting specific permissions, but are considerably more flexible in that they can identify a number of group affiliations for a single user.

The standard UNIX DAC mechanism is permission bits, as is the case with RHEL. However, RHEL implements ACLs as an extended permission mechanism, available at the discretion of the file owner; ACLs are supported only for file system objects.[1]

3.2. Mandatory Access Control

Red Hat Enterprise Linux implements Mandatory Access Control (MAC) through the use of labels maintained by SELinux for processes and objects maintained by the kernel. The MAC policy implements the rule-set based on the Bell-LaPadula model.

Labeled networking as well as labeled printing is provided with the TOE.

3.3. Role-based Access Control

Red Hat Enterprise Linux implements Role-based Access Control (RBAC) through the use of SELinux labels containing role information. A Role-based Access Control policy requires mechanisms whereby the access of users (i.e., subjects) to system resources and data (i.e., objects) can be controlled on the basis of user role and object role.

3.4. I&A

Each user must have a unique identity (i.e., username plus password), and be authenticated prior to obtaining resources and services from the TOE. Note, however, that in a networked environment, user identities are unique to a server, and are neither known globally nor are universally unique. That is, each server maintains its own set of users and their associated passwords and attributes. A user

[1] See Section 6.2.4 of the ST for a fuller discussion of the DAC mechanisms and the algorithm by which access determinations are made.

that has access to more than one server on a network will have a different user identity, and possibly different attributes, on each server for which access is authorized.

Users can change their own passwords. However, an administrator can define the following constraints for the authentication process:

- Maximum duration of a password (i.e., time-to-live);
- Minimum time allowed between password changes;
- Minimum password length;
- Number of days warnings are displayed prior to password expiration;
- Allowed number of consecutive unsuccessful login attempts;
- Disallowed passwords (i.e., the TOE retains a history of recently-used passwords to prevent users from cycling previously-used passwords).

The proper parameters for each of these choices is defined for the evaluated configuration

3.5. Auditing

The TOE audit mechanism allows the generation of audit records for security-related events, and allows the administrator to configure the audit mechanism to collect which events are to be captured and which users are to be audited; it is also possible for the administrator to identify specific users that are not to be audited.

Each audit record contains event-specific information, and identifies whether the request that caused the event was successful or failed, and. An audit record consists of a standard header that includes the following information:

- A unique audit identifier;
- The LoginID of the user who caused the audit record to be generated;
- The Effective User ID of the user at the time the record was generated;
- Date and time the audit record was generated;
- Type of event.

Audit records are stored in ASCII format, and can be searched through the use of the standard UNIX/LINUX *grep* tool.

3.6. Object Reuse

Although the TOE supports several different types of objects, each is managed by the system such that no pre-existing content is provided to users to whom objects are allocated. That is, whenever an object (e.g., buffers, memory extents, disk space) is allocated to a user process, it is managed such that any data that had previously been in the object (i.e., from an earlier process) is unavailable to the new process.

In short, memory pages are initialized to all zeroes when allocated to a process, IPC objects are also initialized to all zeroes, file system objects are created with no content (with the exception of directories and symbolic links).[2]

4. ASSUMPTIONS

4.1. Usage Assumptions

Although there are several assumptions stated in the Security Target[3], the primary conditions are that:

- The TOE is located within controlled facilities and is protected from unauthorized physical access;
- TOE hardware and software are protected from unauthorized modification;
- All authorized users possess authorization for at least some of the data managed on the TOE;
- The TOE operates in a relatively benign environment;
- Unencrypted communications paths, and communications paths within the controlled facility are protected from unauthorized physical access.

4.2. Clarification of Scope

The TOE includes the hardware platform (see Section 8) and all the code that enforces the policies identified (see Section 3). TOE also includes secure communications functions; i.e., SSH V2 and SSL V3).

5. ARCHITECTURAL INFORMATION

The TOE is a multi-user, multi-tasking operating system which can support multiple users simultaneously. A fundamental protection mechanism is the memory management and virtual memory support provided by the hardware. This provides a domain (i.e., supervisor state) in which only the kernel executes.

The TSF comprises two major components: kernel software and trusted processes.

The kernel software executes in supervisor state, which is supported by the memory management mechanism in the hardware. The memory management mechanism insures that only kernel code can execute in the supervisor state (wherein all memory may be accessed), and also serves to protect the kernel code from external tampering. The kernel implements file and I/O services, which provides access to files and devices. The kernel also implements:

- Named pipes

[2] A more complete discussion of object reuse for each of the various object types is contained in Section 6.2.5 of the ST.

[3] See section 3.4 of the ST

- Unnamed pipes
- Signals
- Semaphores
- Shared memory
- Message queues
- Internet domain sockets
- Unix domain sockets.

The trusted processes, which provide the remainder of the TSF, are referred to as “non-kernel TSF” services because they run in user state; they execute in the same hardware domain as user applications. These are protected from external tampering through the process management and memory virtualization mechanisms that implement per-process address spaces, that prevent processes from interfering with each other. They are also protected from unauthorized access by the access control mechanisms of the TSF. The primary non-kernel TSF services are:

- Identification and authentication
- Network application layer services
- Configuration and management commands requiring root privileges.

6. DOCUMENTATION

The TOE is delivered with a combination of hardware and software specific documentation on CD. Hardware specific documentation varies with the model of the TOE. The following software documentation is uniform across TOE hardware platforms:

- Common Criteria EAL4 Evaluated Configuration Guide for Red Hat Enterprise Linux on HP Hardware v2.3 2007-05-31
- Command references for the applications and configuration files implementing security functionality are available as man pages on an installed system

Additional guidance documents are available from Red Hat which have not been assessed by the evaluation. The above mentioned Evaluated Configuration Guide fully and completely explains how to install, configure and administrate the TOE. Moreover, it provides explanations about the intended environment.

Additional man pages to the ones mentioned above are present on the system for applications, configuration files, APIs and others which do not implement security functionality. These man pages have not been reviewed during the evaluation.

7. IT PRODUCT TESTING

7.1. Sponsor Testing

Test configuration

The test results provided by the sponsor were generated on the following systems:

- Intel Xeon (HP DL360)
- Intel Xeon/Pentium (HP Compaq dc7600)
- Intel Xeon EM64T (HP DL360) - dualcore
- Intel Xeon EM64T (HP DL360) - singlecore
- AMD Opteron (HP DL 385) – singlecore
- AMD Opteron (HP DL 385) - dualcore
- AMD Opteron (HP DL 145) - singlecore
- Intel Itanium 2 (rx 3600) – dualcore
- Intel Itanium 2 (rx 2620) – singlecore

The sponsor has performed his tests on the above listed hardware platforms. The software was installed and configured as defined in the Evaluated Configuration Guide [ECG] with additional software packages identified in the Test Plan [TP]. The Test Plan presents the arguments that those additional packages are within the boundary defined by the Security Target and do not constitute a violation of the evaluated configuration (see the chapter headed “Target of Evaluation (TOE) compliance” in [TP]).

The test systems were installed using RHEL5 Server.

Testing approach

The Test Plan provided by the sponsor lists test cases by groups, which reflects the mix of sources for the test cases. The mapping provided lists the TSF/TSFI the test cases are associated with. The Test Plan is focused on the security functions of the TOE and ignores other aspects typically found in developer test plans. The test cases are mapped to the corresponding Functional Specification and HLD.

The sponsor uses several test suites that are integrated into one test system which includes automatic and manual tests to test the TOE.

The LTP test suite is an adapted version of tests from the Linux Testing Project. The LTP tests have a common framework in which individual test cases adhere to a common structure for setup execution and cleanup of tests. Each test case may contain several tests of the same function, stressing different parts (for example, base functionality, behavior with illegal parameters and reaction to missing privileges). Each test within a test case reports PASS respectively OK or FAIL, and the test case summary in batch mode reports PASS if all the tests within the test case passed, otherwise FAIL.

The ACL tests are structured in a very peculiar way. The test cases are comments, shell scripts and expected output. The driver script for the test cases runs the shell commands and compares the output with the expected output in the test scripts. Each output line that matches is tagged with ok,

each line that does not match is tagged with failed. The driver scripts summarize the ok/failed entries and report the number of each of the two flags at the end. The test case reports 101 ok entries when executed successfully. The tests are started in batch mode via the runme shell script.

The OpenSSL tests execute a part of the LTP OpenSSL test suite adapted for the security evaluation.

The audit tests use their own testing framework, where each test is executed twice: once with a positive test goal and once with a negative test goal. The audit tests that do not cover system calls directly but the supporting tools use a similar approach of iterating over the various stages as far as applicable. For each of the areas in the audit test suite, a driver program will perform global setup and run the individual test cases. Results are collected into the log file showing pass or fail verdicts. Additionally, the audit tests also cover MLS logic. By verifying that certain permutations of allowed and denied access requests are audited, the MLS logic is verified as well.

The manual tests cover functionality that can not easily be tested in an automated way, such as serial terminals.

The test results of the sponsor can be found in [TRES]. All the tests were executed successfully (pass/ok) apart from the test cases that are documented to fail or be skipped in the [TP]. The test systems were configured according to the ST and the instructions in [ECG]. The manual test results included in [TRES] also include PASS/FAIL labeling by the sponsor.

The test results provided by the sponsor were generated on the following above mentioned systems.

Testing results

The test results provided by the sponsor were generated on the hardware platforms listed above. As described in the testing approach, the test results of all the automated tests are written to files. In addition a log-file for the LTP tests reports more details on the flow of the tests.

The test results of the few manual tests have been recorded by the sponsor and those results have been presented in separate files.

All test results from all tested platforms show that the expected test results are identical to the actual test results, considering the expected failures stated in the developer’s test plan.

Test coverage

The functional specification has identified the following TSFI:

- system calls
- security critical configuration files (TSF databases)
- trusted programs

- network applications
- virtual files

A mapping provided by the sponsor shows that the tests cover all individual TSFI identified for the TOE. An extension to this mapping developed by the evaluator as documented in the test case coverage analysis document shows that also significant details of the TSFI have been tested with the sponsor's test suite. This therefore satisfies the requirements for the evaluation, since an exhaustive interface specification testing is not required.

Test depth

In addition to the mapping to the functional specification, the sponsor provided a mapping of test cases to subsystems of the high level design and the internal interfaces described in the high level design. This mapping shows that all subsystems the internal interfaces are covered by test cases. To show evidence that the internal interfaces have been called, the sponsor provided the results of test cases that had been executed on a system installed and configured in compliance with the Security Target and the Evaluated Configuration Guide [ECG] but where large parts of the kernel had been compiled with the instrumentation for the gcov coverage analysis tool. This tool allows extracting a profile of all the source code statements that have been executed as part of the tests including also numbers showing how often each source code statement has been executed. Part of the depth analysis was based on the output generated with those gcov instrumented kernels.

Not all of the internal interfaces mentioned in the high-level design could be covered by direct test cases. Some internal interfaces can – due to the restrictions of the evaluated configuration – only be invoked during system startup. This includes especially internal interfaces to load and unload kernel modules, to register /deregister device drivers and install / deinstall interrupt handler. Since the evaluated configuration does not allow to dynamically load and unload device drivers as kernel modules those interfaces are only used during system startup and are therefore implicitly tested there.

7.2. Evaluator Testing

TOE test configuration

The evaluator independently installed the test systems according to the documentation in the Evaluated Configuration Guide [ECG] and the test plan.

rx2620 (Intel Itanium2 based system):

The HP rx2620 is located at the evaluator's facility in Austin, TX. The exact hardware and software configuration of the test system can be found in [TPE] appendix A.1.

The evaluator installed RHEL 5 Server on this system.

Subset size chosen

As the evaluator was integrated in the developer's test team during the development of the test cases and the evaluator's knowledge about the LTP test suite from previous evaluations, the evaluator chose to run the system calls and the libpam test cases out of the newly developed MLS test suite.

Evaluator tests performed

In addition to repeating all the automated developer tests, the evaluator devised tests for a subset of the TOE. The tests are listed in the Evaluator Test Plan [TPE].

The evaluator has chosen these tests for the following reasons:

- The test cases examine some of the security functions of the TOE in more detail than the sponsor supplied test cases. (Object reuse, password quality, access enforcement)
- The test cases cover aspects not included in the developer testing (verification of the long password support, verification of the ACL support in the archival tool, access enforcement on read/write of a file)
- The test cases use a completely different testing approach then the developer tests (MLS rule set verification by analyzing the SELinux policy file).
- As the sponsor-supplied test cases already cover the TOE in a broad sense the evaluator has devised only a small set of test cases.

The evaluator created several test cases for testing a few functional aspects where the sponsor test cases were not considered by the evaluator to be broad enough. During the evaluator coverage analysis of the test cases provided by the sponsor, the evaluator gained confidence in the sponsor testing effort and the depth of test coverage in the sponsor supplied test cases. The analysis has shown a very wide coverage of the TSF, therefore the evaluator devised only a small number of test cases.

Summary of Evaluator test results

The evaluator testing effort consists of two parts. The first one is the re-run of the developer test cases and the second is the execution of the tests created by the evaluator.

The tests were performed at the evaluator's facility in Austin, TX, on the platforms described above. The system was attached to a console and to the evaluator's laptop via the local Ethernet cabling. The RHEL 5 Server operating system with the required patches as well as the test cases and test tools were installed on all three machines by the evaluator according to the instructions in [ECG], [TP] and [TPE]. During the evaluation only the file system type EXT3 with the ACL option and VFAT on the Itanium2 platform for the boot partition were used for hard disk partitions on the test system. The configuration RPM package and the configuration script contained in the rpm ensured the evaluation compliant system configuration. After performing the automated configuration with the configuration script, no further system configuration was performed and only the tools required for testing have been installed. The test systems were therefore configured according to the [ST] and the instructions in the [ECG]. The evaluator used a subset of the automated test cases provided by the sponsor and ran them on the test system via the automated driver scripts according to the test

plan [TP] provided by the sponsor. Manual test were also executed. The log files generated by the test cases were analyzed for completeness and failures.

All the test results conformed to the expected test results from the test plan.

In addition to running the tests that were provided by the sponsor according to the test plan from the sponsor, the evaluator decided to run some additional test cases on the provided test systems:

- Password Quality Tests
  This test was performed to verify that the password quality settings prevent trivial passwords. See [TPE], section 3.1
- Verification of the use of MD5 passwords
  This test was performed to verify that long passwords can be used on the TOE due to the MD5 algorithm used for storing the passwords instead of using the classic crypt function that truncates passwords at eight characters. See [TPE], section 3.2
- Verification the SUID programs do not change the real UID
  This test was performed to verify that SUID programs do not change the real UID, only the effective UID. See [TPE], section 3.3.
- Testing of object reuse in regular file system objects
  This test checks for object reuse in regular files by creating a large spares file and trying to find non-zero data in the spares area. See [TPE], section 3.4.
- Check for data import / export with DAC enforcement
  Although no claims in the ST are made about data import and export, the evaluator deemed it necessary to check for the correct functioning of the star utility mentioned for this purpose in the Evaluated Configuration Guide. By testing this utility the evaluator also had a simple ACL enforcement test. See [TPE], section 3.5 for the test case.
- Test for disabling the password suggestion
  This test verifies that no password suggestions are presented during changing the password after a specific configuration to verify that the suggestions are done by a particular PAM module. See [TPE], section 3.6.
- Test for access check enforcement
  This test verifies that the decision made during open() system call (e.g. open the file as read-only) is enforced during read/write system calls. See [TPE], section 3.7.
- Test for LD_LIBRARY_PATH
  This test checks that the environment variable LD_LIBRARY_PATH is unset when calling a SUID/SGID application.
- Validation of Bell-LaPadula logic
  This test case demonstrates whether the SELinux policy adheres to Bell-LaPadula. See [TPE], section 3.9.
- Test for MLS override attributes
  This test case demonstrates whether all applications granting MLS override capabilities to user_r are part of the FSP mapping table. See [TPE], section 3.10.

- Test for trusted objects
  This test case demonstrates whether no ranged object is provided that allows storage or transmission of user data. See [TPE], section 3.11.
- Test for ranged objects
  This test case demonstrates whether no ranged object is provided that allows storage or transmission of user data. See [TPE], section 3.12.

All tests passed successfully.

8. EVALUATED CONFIGURATION[4]

The evaluated configurations are:

- HP Intel Itanium2 (single and multi-core) processor based servers:
  - HP Integrity Superdome product line
  - HP Integrity rx product line
  - HP Integrity cx product line
  - HP Integrity BL product line
- Intel Xeon based servers with EM64T 64bit extensions (single and multi-core), and HP AMD Opteron processor (single and multi-core):
  - HP Proliant ML product line (EM64T capable models)
  - HP Proliant DL product line (EM64T capable or Opteron models)
  - HP ProLiant BL product line (EM64T capable or Opteron models)
- HP Intel Pentium and Xeon processor based servers without EM64T extensions:
  - HP Proliant ML product line (except EM64T capable models)
  - HP Proliant DL product line (except EM64T capable or Opteron models)
  - HP Proliant BL product line (except EM64T capable or Opteron models)
- HP Intel Xeon processor based systems:
  - HP xw product line
- HP Intel Pentium 4 processor based systems:
  - HP xw product line
  - HP Compaq dc series product line

[4] For more complete information on the evaluated configurations, see Section 2.4 of the Security Target.

9. RESULTS OF THE EVALUATION[5]

The evaluation team determined the product to be **CC Part 2 extended, CC Part 3 conformant, CAPP/LSPP/RBAC conformant,** and to meet the requirements of **EAL 4 augmented by ALC_FLR.3**. In short, the product satisfies the security technical requirements specified in *HP Red Hat Enterprise Linux Version5 Security Target for CAPP, LSPP and RBAC compliance,* Version 3.9, 2007-05-31.

10. VALIDATOR COMMENTS

The Validator determined that the evaluation and all of its activities were performed in accordance with the CC, the CEM and CCEVS practices.

The Validator has the following observation:

- While the TOE distribution media includes a Graphical User Interface (GUI), it is not installed by default, is not part of the Evaluated Configuration and was not evaluated.

11. SECURITY TARGET

The ST, *HP Red Hat Enterprise Linux Version5 Security Target for CAPP, LSPP and RBAC compliance,* Version 3.9, 2007-05-31 is included here by reference.

[5] The terminology in this section is defined in CC Interpretation 008, specifying new language for CC Part 1, section/Clause 5.4.

12. LIST OF ACRYONYMS

CC	Common Criteria
CCEVS	Common Criteria Evaluation and Validation Scheme
CCTL	Common Evaluation Testing Laboratory
CEM	Common Evaluation Methodology
EAL	Evaluation Assurance Level
ETR	Evaluation Technical Report
NIAP	National Information Assurance Partnership
NIST	National Institute of Standards & Technology
NSA	National Security Agency
PP	Protection Profile
ST	Security Target
SMP	Symmetric Multiprocessing
TOE	Target of Evaluation
TSF	TOE Security Function
TSFI	TOE Security Function Interface
UP	Uniprocessor

13. BIBLIOGRAPHY

[1] Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and general model, Version 2.3.

[2] Common Criteria for Information Technology Security Evaluation – Part 2: Security functional requirements, Version 2.3.

[3] Common Criteria for Information Technology Security Evaluation – Part 3: Security assurance requirements, Version 2.3.

[4] Common Evaluation Methodology for Information Technology Security – Part 1: Introduction and general model, version 2.3.

[5] Common Evaluation Methodology for Information Technology Security – Part 2: Evaluation Methodology, version 2.3.